# Constrained Versions of Some Natural NP-Complete Problems

*By*

**KALLA DURGA MALLESWAR**



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

APRIL, 1993

# Constrained Versions of Some Natural $\mathcal{NP}$-Complete Problems

*A Thesis Submitted*

*in Partial Fulfilment of the Requirements*

*for the Degree of*

MASTER OF TECHNOLOGY

*by*

**Kalla Durga Malleswar**

to the

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**INDIAN INSTITUTE OF TECHNOLOGY, KANPUR**

April, 1993

# CERTIFICATE

This is to certify that the work contained in this thesis titled **"Constrained Versions of Some Natural $NP$-Complete Problems"**, was carried out under my supervision by **Kalla Durga Malleswar** and it has not been submitted elsewhere for a degree.

**(Somenath Biswas)**
Dept. of Comp. Sc. and Engg.
I.I.T., Kanpur
April 1993

*To My Parents - Two Wonderful People*

## ACKNOWLEDGEMENT

I express my sincerest thanks and gratitude to my guide, Dr. Somenath Biswas, who has been an inestimable help through the course of my thesis. I am greatly indebted to him for enriching my knowledge. I would like to thank Dr. Ahuja and Dr. M. Agrawal for discussions that helped in the development of this work.

I thank all my friends and classmates who have made my stay here really enjoyable. My thanks to the staff of CSE library and central library for being so helpful.

MALLESWAR

April, 1993.

## Abstract

In this thesis we examine *subgraph* $NP$-complete problems in terms of the constraints satisfied by the solution *subgraph*. Towards the characterization of natural properties that make a problem hard we have considered two kinds, namely, local and global constraints satisfied by the solution subgraph. We show that certain combinations of these constraints make a problem intractable and certain other combinations make the problem easy. In particular, we prove that $R$-regular subgraph problem becomes $NP$-Complete when the lower bound on the number of components or the upper bound on the number of components or an upper bound on the number of nodes in each component is specified.

# Contents

# Chapter 1

# Introduction

What makes a problem hard? what makes a problem easy? It is now possible to sharply focus these fundamental concerns of computing in terms of certain formal notions that capture the intuitive understanding of the concepts "hard" and "easy". Both from theoretical, as well as from empirical and pragmatic considerations, it is now a consensus to equate the concept of "hard" or "intractable" with $NP$-Completeness and to equate "easy" or " well-solved" with polynomial time solvability. We will, therefore, understand what makes a problem hard if we can discover properties, in terms of natural parameters of problem domains, which are necessary and sufficient to capture $NP$-Completeness.

That is, no doubt, a deep and a difficult task calling for the formulation of a metatheory. What we do in this thesis is to prove some results which may be used as data for building the metatheory. To discover natural properties that will explain the $NP$-Completeness phenomenon, it is clearly of interest to examine the boundary between provably polynomial time and $NP$-Completeness.

Problems analyzed for $NP$-Completeness are often distilled from less elegant applied problems. So even when a problem is proven to be $NP$-complete, the applied problem can still be solvable in *polynomial* time as some of the details that were dropped in the distillation process might alter the problem enough to make it easy. If not, there still might be significant special cases that can be solved in *polynomial* time. A subproblem is obtained whenever additional restrictions are placed on the allowed instances. Even though a problem is $NP$-complete, each of the subproblems

of it might independently be either $NP$-complete or p time solvable. Thus the subproblems of any $NP$-complete problem can be viewed as lying on different sides of the boundary between $P$ time solvability and intractability. The goal in analyzing a problem is to determine which subproblems lie on each side. Further, problems can be changed by making the given problems more-restricted or less-restricted.

Problems from graph theory are quite common among $NP$-Complete problems. In a *subgraph* problem the solution of the given graph $G$, is required to be a *subgraph* of $G$. Subgraph problems were earlier classified as node deletion $NP$-Complete problems and edge deletion $NP$-Complete problems. We take a somewhat different view. Any *subgraph* problem can be viewed as an instance of graph which requires a *subgraph* satisfying one or more constraints. Further, these constraints can be classified as *local* or *global*. *Local* constraints such as constraints on degrees are satisfied locally, at limited contexts. *Global* constraints are much diverse in nature like, the number of components in the *subgraph*, number of nodes in each component, total weight of the *subgraph*, etc. We examine *subgraph* $NP$-complete problems in terms of the constraints satisfied by the solution *subgraph*.

## 1.1 Preliminaries

A problem is a general question to be answered, usually possessing several parameters, or free variables, whose values are left unspecified. A problem is described by giving: (1) a general description of all its parameters, and (2) a statement of what properties the answer, or solution, is required to satisfy. An instance of a problem is obtained by specifying particular values for all the problem parameters. Algorithms are general, step-by-step procedures for solving problems. An algorithm is said to solve a problem P if that algorithm can be applied to any instance I of P and is guranteed always to produce a solution for that instance I. A function f(n) is $O(g\ (n))$ whenever there exists a constant c such that $|f(n)| \leq c.|g(n)|$ for all values of $n \geq 0$. A *polynomial* time algorithm is defined to be one whose time complexity function is $O(p\ (n))$ for some *polynomial* function p, where n is used to denote the input length. There is wide agreement that a problem has not been 'well-solved' until a *polynomial* time

algorithm is known for it. Hence a problem shall be referred to as intractable if it is so hard that no *polynomial* time algorithm can possibly solve it. We distinguish between two different causes of intractability. The first is that the problem is so difficult that an exponential amount of time is needed to discover a solution. The second is that the solution itself is required to be so extensive that it cannot be described with an expression having length bounded by a *polynomial* function of the input length. The existence of intractability due to the second cause is apparent from the problem definition in most cases. This type of intractability can be regarded as a signal that the problem is not defined realistically, because we are asking for more information than we could ever hope to use. Thus the attension is restricted to the first type of intractability. Accordingly, only problems for which the solution length is bounded by a *polynomial* function of the input length will be considered. Most of the apparently intractable problems encountered in practice are decidable and can be solved in *polynomial* time with the aid of a nondeterministic computer. Thus, none of the proof techniques developed so far is powerful enough to verify the apparent intractability of these problems. Hence, efforts were also focussed on learning more about the ways in which various problems are interrelated with respect to their difficulty. The principal technique used for demonstrating that two problems are related is that 'reducing' one to the other, by giving a constructive transformation that maps any instance of the first problem into an equivalent instance of the second. Such a transformation provides means for converting any algorithm that solves the second problem into a corresponding algorithm for solving the first problem.

In his breakthrough paper Stephen Cook [C71], focused attention on the class NP of decision problems that can be solved in polynomial time by a nondeterministic computer. He proved that one particular problem in NP, called the 'Satisfiability' problem, has the property that every other problem in NP can be polynomially reduced to it. If the 'Satisfiability' problem can be solved with a polynomial time algorithm, then so can every problem in NP, and if any problem in NP is intractable, then the 'Satisfiability' problem also must be intractable. Thus, in a sense, the satisfiability problem is the hardest problem in NP. Since then a wide variety of other

problems have been proved equivalent in difficulty to this problem, and this equivalence class, consisting of the 'hardest' problems in NP, has been given a name: the class of $NP$-Complete problems. In the next section we will list some of the famous NP-Complete problems[GJ79] referred to elsewhere in this thesis.

## 1.2 Some Basic $NP$-Complete Problems

### 1.2.1 Satisfiabiality Problem

Instance: A set $U$ of variables and a collection $C$ of clauses over $U$.

Question: Is there a satisfying truth assignment for $C$?

### 1.2.2 Hamiltonian Circuit

Instance: A graph $G$=($V$,$E$).

Question: Does $G$ contain a simple circuit that includes all the vertices of $G$ ?

### 1.2.3 3-Satisfiability (3SAT)

Instance: Collection $C = \{c_1, c_2, ..., c_m\}$ of clauses on a finite set $U$ of variables such that $|c_i|= 3$ for $1 \leq i \leq m$.

Question: Is there a truth assignment for $U$ that satisfies all the clauses in C?

### 1.2.4 3-Dimensional Matching(3DM)

Instance: A set $M \subseteq W \times X \times Y$, where $W$,$X$,and $Y$ are disjoint sets having the same number q of elements.

Question: Does $M$ contain a matching, that is, a subset $M' \subseteq M$ such that $| M' |= q$ and no two elements of $M'$ agree in any coordinate?

### 1.2.5 Vertex Cover(VC)

Instance: A graph $G$=($V$,$E$) and a positive integer $K \leq |V|$.

Question: Is there a vertex cover of size $K$ or less for $G$, that is, a subset $V' \leq V$ such that $| V' |\leq K$ and, for each edge {u,v} $\in E$, at least one of u and v belong to $V$?

### 1.2.6 Min. Max. flow spanning tree

Instance: Graph $G$=($V$,$E$), bound $B > 0$.

Question: Is there a spanning tree $T$= ($V$,$E'$) for $G$ such that for all edges e $\in E'$, the flow on e , i.e. the number f(e) of pairs of vertices u,v $\in V$ that would be disconnected if e were removed from $T$, is no more than $B$?

### 1.2.7 Bounded diameter decomposition

Instance: Graph $G$=($V$,$E$), bound $D > 0$.

Question: Is there a subset $E' \subseteq E$ such that both $G'$=($V$,$E'$) and $G''$=($V$,$E - E'$) have a diameter $D$, i.e. for every u,v $\in V$ there is a path joining u and v of length $D$, or less ?

### 1.2.8 Partition into triangles

Instance: Graph $G = (V, E)$, with $|V| = 3q$ for a positive integer q.

Question: Is there a partition of $V$ into q disjoint sets $V_1, V_2, \ldots, V_q$ of three vertices each such that, for each $V_i = \{v_{i1}, v_{i2}, v_{i3}\}$, the three edges $\{v_{i1}, v_{i2}\}, \{v_{i2}, v_{i3}\}$, and $\{v_{i3}, v_{i1}\}$ all belong to $E$?

### 1.2.9 Clique

Instance: A graph $G = (V, E)$ and a positive integer $j \leq V$.

Question: Does $G$ contain a clique of size j or more, that is, a subset $V' \subseteq V$ such that $| V' | \geq j$ and every two vertices in $V'$ are joined by an edge in $E$?

### 1.2.10 Exact Cover by 3-Sets(X3c)

Instance: A finite set $X$ with $| X |$ = 3q and a collection $C$ of 3-element subsets of $X$.

Question: Does $C$ contain an exact cover for $X$, that is, a subcollection $C' \subseteq C$ such that every element of $X$ occurs in exactly one member of $C'$?

## 1.3 PTime - $NP$-Completeness Boundary: Known Results

This section contains some of the latest results concerning the PTime - $NP$-Completeness boundary for various graph problems [J84],[J85].

### 1.3.1 Hamiltonian Circuit

Hamiltonian Path is $NP$-Complete for reducible flow graphs, edge graphs. *Polynomial* time solvable subcases. Series-parallel graphs, proper interval graphs(interval graphs corresponding to collections of intervals, no one of which is completely contained in another), proper circular arc graphs that are not proper interval graphs, 4-connected planar graphs.

### 1.3.2 Partition into triangles

$NP$-Complete for planar graphs and hence the problem of finding a collection of vertex disjoint triangles that cover a maximum number of vertices in a planar graph $G$ is $NP$-Hard. However if one wants a collection of vertex disjoint $K_3$'s and $K_2$'s rather than just $K_3$'s, this problem can be solved in $P$-time, even for non-planar graphs.

### 1.3.3 Edge partition into triangles

Remains $NP$-Complete even if there is an edge partition into triangles for the complement of $G$.

### 1.3.4 Dominating set

Dominating set problem is $NP$-Complete even for 3-regular planar graphs.But, it is solvable in P for strongly chordal graphs, circular arc graphs, cacti, series parallel graphs and for graphs that differ from trees by only a fixed number of edges. For trees even k-dominating set problem, where every vertex must be with in the distance k of some vertex in the dominating set can be solved in P. Unlike the closely related vertex cover Problem, Dominating set is $NP$-Complete for bipartite graphs and chordal

graphs. Connected dominating set is $NP$-Complete for graphs that are (a) 4-Regular graphs (not- necessarily planar) (b) planar graphs with maximum degree 4. Yet another variant is the one in which the dominating set is required to induce a cycle. This version is $NP$-Complete for planar 2-connected graphs but solvable in *polynomial* time for outer planar graphs.

### 1.3.5 Chromatic Index

Remains $NP$-Complete even if chromatic index of the complement of $G$ equals the obvious Lower bound of maximum vertex degree. Solvable in $P$ for outerplanar graphs and for planar graphs with a vertex of degree 8 or more. The *polynomial* solvable case of bipartite graphs becomes $NP$-Complete when generalized by adding to the instance a partition of the edges and requesting that no two edges in the same subset of the partition be assigned the same color.

### 1.3.6 Min. Max. flow spanning tree

If instead of asking that the maximum of f(e)'s be less than bound $B$ we ask that their sums be less than $B$, we have the $NP$-Complete problem the shortest total path length spanning tree problem and a special case of optimum communication spanning tree. If for each edge we count only the number of vertices it seperates from a prespecified root $V_o$, and we wish that the maximum of these revised flows be bounded by $B$, then we get an $NP$-Complete subproblem of capacitated spanning tree. If instead of asking that the maximum of f(e)'s be less than bound $B$ we ask that their sums be less than $B$, we have the $NP$-Complete problem the shortest total path length spanning tree problem and a special case of optimum communication spanning tree. If for each edge we count only the number of vertices it seperates from a prespecified root $V_o$, and we wish that the maximum of these revised flows be bounded by $B$, then we get an $NP$-Complete subproblem of capacitated spanning tree. However, if we replace maximum by sum in this variant we obtain *polynomial* time solvability, since all we are asking for is a shortest path tree. If instead of asking that all flows be $B$ or less ,we ask that there exist atleast one edge with f(e) greater

than $B$, the problem becomes $NP$-Complete in the unrooted case and polynomail time solvable in the rooted case. For the rooted case if one wishes simultaneously to minimize the tree height, minimize the tree diameter, minimize the sum of the distances to the root or maximize the maximum vertex degree , each of these doubly constrained problems is $NP$-Hard, despite the fact that the individual constraints can be handled in *polynomial* time.

### 1.3.7 Max. Weight connected subgraph

This problem remains $NP$-Complete even if $G$ is planar with maximum degree 3 and all weights either +1 or -1. Also $NP$-Complete is the variant in which the labels are on the edges rather than the vertices, and the desired subgraph is required to be a subtree.

### 1.3.8 Bounded Diameter Decomposition

This remains $NP$-Complete even if $D$= 3 and $G$ is bipartite. Analogous results hold for radii and for directed graphs. The two constraint problem of telling whether a given graph $G$ has a subgraph with diameter atmost $D$ and max. vertax degree at most k is clearly $NP$-Complete. Indeed the degree bound alone is enough to ensure the $NP$-Completeness, so long as one desires a connected subgraph. On the other hand, if all one wants is a not-necessarily connected subgraph of max. edge weight that obeys an upper bound on vertex degrees( but no diameter bound),we have the b-matching problem of Edmonds, and maximum weight subgraphs can be found in *polynomial* time even if seperate upper bounds are specified for each vertex, and even if lower bounds are also provided. If the upper and lower bounds are equal and $G$ is a tree one can even compute the number of such graphs in *polynomial* time.

# Chapter 2

# Constrained Subgraph Problems

Let us first impose only one constraint on the solution *subgraph*, namely the *local* constraint 'degree'. This defines the degree constrained *subgraph* problem.

Instance: Undirected graph $G = (V, E)$, Where '$V$' is the set of vertices and '$E$' the set of edges between them; For each vertex $v \in V$ a set $D_v$ of allowable degrees. Question: Does $G$ contain a *subgraph* $G' = (V, E')$ where $E' \subset E$ such that if d(v) is the degree of vertex $v \in V$ in $G' \forall v \in V$, then $d(v) \in D_v, \forall v \in V$ ?

**Theorem :** Degree Constrained *subgraph* problem (DCSP) is $NP$-Complete.

**Proof:** (Transformation from Exact Cover By 3-Sets).

Exact Cover By 3-Sets (XS3): Given a family $S = \{c_1, ..., c_s\}$ of three element subsets of a set $T = \{t_1, ..., t_{3m}\}$, Does there exist a sub family $S' \subseteq S$ of pairwise disjoint sets such that $\bigcup c = T$ ?

We will show that XS3 is reducible to DCSP.

Given any instance of XS3 we define an instance of DCSP as follows.

$V = S \cup T$, Where $S = \{c_i / c_i \in S\}$,

$T = \{t_i / t_i \in T\}$,

$E = \{$ (c,t): t $\in c \in S$ $\}$.

$D_v = \{\{0, 3\}$ if $v \in S$ and $\{1\}$ if $v \in T\}$

Now, the graph $G$= ($V$,$E$) will have a *subgraph* with the desired degree constraints if and only if the given XS3 instance has an exact cover.

When the given XS3 has a solution, it should comprise of 'm' three element subsets which should cover the '3m' elements, the corresponding DCSP will have a solution

where the vertices corresponding to chosen three element subsets will have degree three and other subsets will have degree zero and since this choice of subsets covers the set '$T$', all $t_i$ nodes will have degree one. It can be seen that when there is a solution to DCSP , since $t_i$'s are connected only to $c_i$'s and should have degree one, only $mc_i$'s will have degree '3'(others having zero) which exactly cover the set '$T$'.

Now, the reduction is shown to work and the transformation is *polynomial* time computable and since XS3 is $NP$-Complete it follows that DCSP is $NP$-Complete.

## 2.1 Single Degree Constrained Subgraph Problem

Consider the case where each set $D_v$ should contain only one integer, i.e. we are interested in a *subgraph* which has exactly '$b_i$' edges associated with node '$v_i$'. As we will soon see this problem can be solved in *Polynomial* time using 'Matching' techniques.

Problem: Find a *subgraph* of an Undirected graph $G = (V, E)$ where each node '$v_i$' is associated with an integer '$b_i$' such that in the *subgraph* each node has exactly $b_i$ edges.

Algorithm: We shall give a transformation that converts an instance of degree constrained matching (b-matching) of $G$ to an instance of ordinary matching problem on graph $G'$ derived from $G$[GM84]. Later we can apply complete matching algo. on $G'$ to get the degree constrained matching of $G$.

Let the given graph $G = (V, E)$, where $| V |= n$ and the vertices are $v_1, ...v_n$. In linear time we can check that $\forall v_i degree(v_i) \geq b_i$. The Graph $G'$ is defined as follows. For each vertex $v_i of G$ associate two disjoint sets of vertices of $G'$: $A_i$ (cardinality of $A_i = Degree(v_i)$ ), where each element of '$a_{iu}$' of $A_i$ corresponds to an edge 'u' of $G$ incident with $v_i$; $C_i$ (cardinality of $C_i = Degree(v_i) - b_i$), where the elements of $C_i$ are denoted by $C_{ir}(r = 1, \ldots, | C_i |)$. The set of vertices of $G'$ is thus the union of all $A_i$ and of all $C_i$(for $v_i$ in $V$). The edges of $G'$ are then obtained, on the one hand, for i=1,...,n, by linking every vertex of $A_i$ with every vertex of $C_i$, and on the other hand, for every edge $u = (v_i, v_j)$ of $G$, by linking the vertices $a_{iu}$ and $a_{ju}$ to make up the edge $u' = (a_{iu}, a_{ju})$ of $G'$. If $K'$ is a matching of $G'$ which saturates all vertices, then the set of edges $u = (v_i, v_j)$ such that $u' = (a_{iu}, a_{ju}) \in K'$ forms a b-matching of $G$, and conversely. Now, when all $c_{ir}(i = 1...n)$ are saturated, some of the edges $u' of G'$ corresponding to the actual edges $u = \{v_i, v_j\}$ in $G$ are disabled as $c_{ir}$'s are connected only to $a_{iu}$'s. Hence the matching in $G'$ can contain at most '$b_i$' edges with vertex $v_i$ in $G$ and since we search for a complete matching in $G'$ it follows that *subgraph* will contain exactly $b_i$ edges associated with $v_i$.

## 2.2 Single Degree and Components Constrained Subgraph Problem

For the SDCSP problem matching techniques do not seem to work if we specify a bound '$K$' on the no. of components in the *subgraph.* The following theorems strengthen this notion.

Instance: Undirected graph $G = (V, E)$, Where '$V$' is the set of vertices and '$E$' the set of edges between them; For each vertex $v_i \in V$ an integer $b_i$ that specifies the degree bound of node $v_i$ and an integer $K \leq \mid V \mid$, that fixes the lower bound on the no. of components in the *subgraph.*

Question: Does $G$ contain a *subgraph* $G' = (V, E')$ where $E' \subseteq E$ such that each vertex $v_i \in V$ is associated with exactly $b_i$ edges and the number of components in the *subgraph* is greater than or equal to $K$?

**Theorem:** SDCCSP is $NP$-Complete.

**Proof:** (Transformation from Exact Cover By 3-Sets).

Exact Cover By 3-Sets (XS3): Given a family $S = \{c_1, ..., c_s\}$ of three element subsets of a set $T = \{t_1, ..., t_{3m}\}$, Does there exist a sub family $S' \subseteq S$ of pairwise disjoint sets such that $\bigcup c = T$ ?

We will show that XS3 is reducible to SDCCSP.

Given any instance of XS3 we define an instance of SDCCSP as follows (Ref. Fig. 1).

$V = S \cup T \cup P$, Where $S = \{c_i / c_i \in S\}$, $\mid S \mid= n$.

$T = \{t_i / t_i \in T\}$, $\mid T \mid= 3m$.

$P = \{p_{ij} / (i = 1..., n - m)(j = 1, 2, 3)\}$, $\mid P \mid= 3(n - m)$.

$E = ST \cup SP \cup PP$, Where $ST = \{(c, t) : t \in c \in S\}$

$SP = \{(c, p) : c \in S, p \in S\}$

$PP = \{(p_{i1}, p_{i2}), (p_{i2}, p_{i3}), (p_{i3}, p_{i1}) : p_i \in P\}$.

/* $PP : p_{i1}, p_{i2}, p_{i3}$ form a triangle. Hence n-m triangles */

/* $SP : n - m$ triangles connected to all c nodes */

$b_i = 1$; if $v_i \in T$

3; if $v_i \in P$

3; if $v_i \in S$

$K = n$.

Now, the graph $G = (V, E)$ will have a *subgraph* with the desired degree constraints and the number of components $\geq n$, if and only if the given XS3 instance has an exact cover.

Let us first see that the maximum possible number of components is 'n'. Because, whenever a node has degree requirement **3** the component in which this node is present should have at least 4 (including this one) nodes. In our instances we have '4n-3m' nodes requiring 3 and '3m' nodes requiring 1. As nodes requiring 1 are not connected among themselves they should all be connected to degree 3 nodes. The total nodes being 4n we can maximum get n components. So even though constraint requires only greater than or equal to $K$ it amounts to saying equal to $K$.

When the given XS3 has a solution, it should comprise of 'm' three element subsets which should cover the '3m' elements, the corresponding SDCCSP will have a solution where the vertices corresponding to chosen three element subsets will be connected to the three elements present in them , thus contributing 'm' components, other subsets will be connected to one triangle each contributing 'n-m' components. since this choice of subsets covers the set '$T$' all $t_i$ nodes will have degree one and the total components in the *subgraph* will be 'n'. It can be seen that when there is a solution to SDCCSP, since $t_i$'s are connected only to $c_i$'s and should have degree one, only $mc_i$'s will have degree '3' as a result remaining 'n-m' $c_i$'s are forced to have connections with the triangles, each $c_i$ connected to all the vertices of a triangle. if that is not the case, since the degree requirement of 't' nodes is '1' they will be connected to say 'l' clauses satisfying a total '3m' degree requirement of them , if l¿m there still remains 3(l-m) degree requirement which can only be satisfied by connections to nodes corresponding to triangles. In that case those $c_i$'s whose degree requirements are 1 or 2 can not be connected to all the three vertices of triangles, so if all the edges (or at least two) of triangle are present in any solution then 2 or 1 remaining nodes should be connected

to some other subset vertex in which case these subset vertices, all the $t_i$'s connected to these subset vertices and the nodes corresponding to the triangle belong to the same component decreasing the max. possible components by 1 and there is no other way to compensate this. Similar argument holds for other cases. The only way to get $K$ components is to connect $t_i$'s in groups of three to one $c_i$ and the remaining $c_i$'s to one triangle each. Hence the given XS3 will have an exact cover.

Now, the reduction is shown to work and the transformation is *polynomial* time computable and since XS3 is $NP$-Complete it follows that SDCCSP is $NP$-Complete.

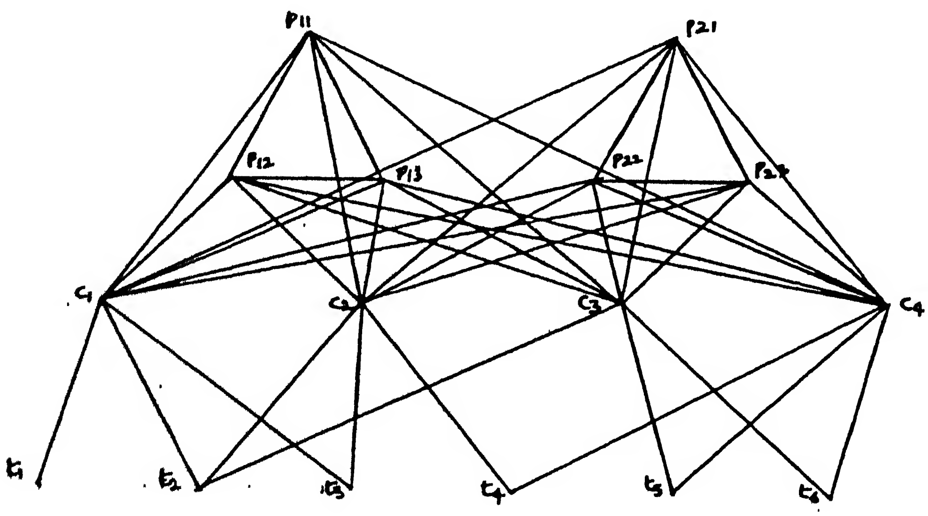


**Fig.1. Reduction for SDCCSP**

## 2.3 Connected Single Degree Constrained Subgraph Problem

The no. of components constraint will make this problem intractable even when the bound is the upper limit.

Instance: Undirected graph $G = (V, E)$, Where '$V$' is the set of vertices and '$E$' the set of edges between them; For each vertex $v_i \in V$ an integer $b_i$ that specifies the degree bound of node $v_i$.

Question: Does $G$ contain a connected *subgraph* $G' = (V, E')$ where $E' \subseteq E$ such that each vertex $v_i \in V$ is associated with exactly $b_i$ edges?

**Theorem:** Connected Single Degree Constrained Subgraph Problem is $NP$-Complete.

**Proof:** (Transformation from Hamiltonian Cycle Problem).

Hamiltonian Cycle Problem: Given an Undirected graph $G$= ($V$,$E$), Where '$V$' is the set of vertices and '$E$' the set of edges between them, Does there exist a cycle which visits each node exactly once?

We will show that HCP is reducible to SDCCSP.

Given any instance of HCP we define an instance of SDCCSP as follows.

Let the given graph $G$=($V$,$E$), where $\mid V \mid= n$ and the vertices are $v_1, ...v_n$. The Instance of SDCCSP is defined as follows: It contains the same graph $G$ and only the vector $B$ is created.

$b_i : 2, \forall v_i \in V.$

As can be seen a connected subgraph with all nodes having degree exactly two is nothing but a Hamiltonian Cycle. Hence, it follows that SDCCSP is $NP$-Complete.

## 2.4 Regular and Components Constrained Subgraph Problems

Relaxing the Degree Constraint in such a way that all nodes should have the same degree requirement i.e. a *regular subgraph* , would not reduce the complexity of the problem significantly. We prove that $R$-Regular Subgraph Problem becomes $NP$-Complete when we specify the lower bound on the number of components or the upper bound on the number of components or an upper bound on the number of nodes in each component. The following theorems strengthen this notion.

Instance: Undirected graph $G = (V, E)$, Where '$V$' is the set of vertices and '$E$' the set of edges between them and an integer $K \leq |V|$, that fixes the lower bound on the no. of components in the *subgraph.*

Question: Does $G$ contain a 3-Regular subgraph $G' = (V, E')$ where $E' \subseteq E$ such that the number of components in the *subgraph* is greater than or equal to $K$?

**Theorem:** 3-RCCSP is $NP$-Complete.

**Proof:** (Transformation from Exact Cover By 3-Sets).

Exact Cover By 3-Sets (XS3): Given a family $S = \{c_1, ..., c_s\}$ of three element subsets of a set $T = \{t_1, ..., t_{3m}\}$, Does there exist a sub family $S' \subseteq S$ of pairwise disjoint sets such that $\bigcup c = T$ ?

We will show that XS3 is reducible to 3-RCCSP.

Given any instance of XS3 we define an instance of RCCSP as follows (Ref. Fig. 2).

$V = S \cup T \cup P \cup H$, Where $S = \{c_i / c_i \in S\}, |\, S| \,| = n$.

$T = \{t_i / t_i \in T\}, \mid T \mid = 3m$.

$P = \{p_{ij} / (i = 1..., n - m)(j = 1, 2, 3)\}, \mid P \mid = 3(n - m)$.

$H = \{h_i\}, \mid H \mid = 6m$.

$E = ST \cup SP \cup PP \cup HH \cup TH$, Where $ST = \{(c, t) : t \in c \in S\}$

$SP = \{(c, p) : c \in S, p \in S\}$

$PP = \{(p_{i1}, p_{i2}), (p_{i2}, p_{i3}), (p_{i3}, p_{i1}) : p_i \in P\}$.

$HH = \{(h_i, h_{(i+1)}), (h_{6m}, h_1) : h_i \in H\}$.

$TH = \{(t_i, h_i), (t_i, h_{(i+3m)}) : t_i \in T, h_i \in H\}$

/* $PP$ : $p_{i1}, p_{i2}, p_{i3}$ form a triangle. Hence n-m triangles */

/* $SP$: n-m triangles connected to all c nodes */

/* $HH$: All $h_i$'s are connected in circular fashion. Hence a cycle of length 6m */

/* $TH$: Each $t_i$ is connected to two nodes in the cycle formed by $HH$ */

$K = n - m + 1$.

Now, the graph $G = (V, E)$ will have a 3-regular subgraph with number of components $\geq$ n, if and only if the given XS3 instance has an exact cover.

Let us first see that the maximum possible number of components is 'n-m+1'. Because, whenever a node has degree requirement 3 the component in which this node is present should have at least 4 (including this one) nodes. In our instances we have a set of nodes (6m) corresponding to '$H$' are connected in a cycle and there are no other connections among these nodes except that each $h_i$ is connected to a $t_i$(or $t_{(i-3m)}$). Since we require a 3-regular graph all the edges that have a $h_i$ node as an end point should be present in the desired *subgraph*. As $t_i$ nodes are not connected among themselves they should all be connected to $c_i$ nodes. The total nodes(barring $H$) being 4n we can maximum get 'n' components. Of these the m components corresponding to the 3m$t_i$ nodes and m$c_i$ nodes are connected to each other via the $H$ cycle which reduces the no. of components by m-1. So even though constraint requires only greater than or equal to $K$ the only possible number is $K$.

When the given XS3 has a solution, it should comprise of 'm' three element subsets which should cover the '3m' elements, the corresponding 3-RCCSP will have a solution where the vertices corresponding to chosen three element subsets will be connected to the three elements present in them and the all the $t_i$ nodes are connected to the $H$ cycle which contributes one component to the solution *subgraph*, other subsets will be connected to one triangle each contributing 'n-m' components. Thus the total components in the subgraph will be 'n-m+1'. It can be seen that when there is a solution to 3-RCCSP, since $t_i$'s are connected to $H$ cycle in any solution and at least m$c_i$'s should be connected to $t_i$'s all these 10m (6m+3m+m) nodes will be in one component. There will be 4(n-m) nodes left which can yield maximum components

i.e. n-m, when connected in groups of four. Because of the connections the only way to have got it is by connecting each $c_i$ to a separate triangle. Thus the m$c_i$ nodes connected to the $t_i$ nodes correspond to the exact cover of the given XS3 instance.

Now, the reduction is shown to work and the transformation is *polynomial* time computable and since XS3 is $NP$-Complete 3-RCCSP will also be $NP$-Complete.

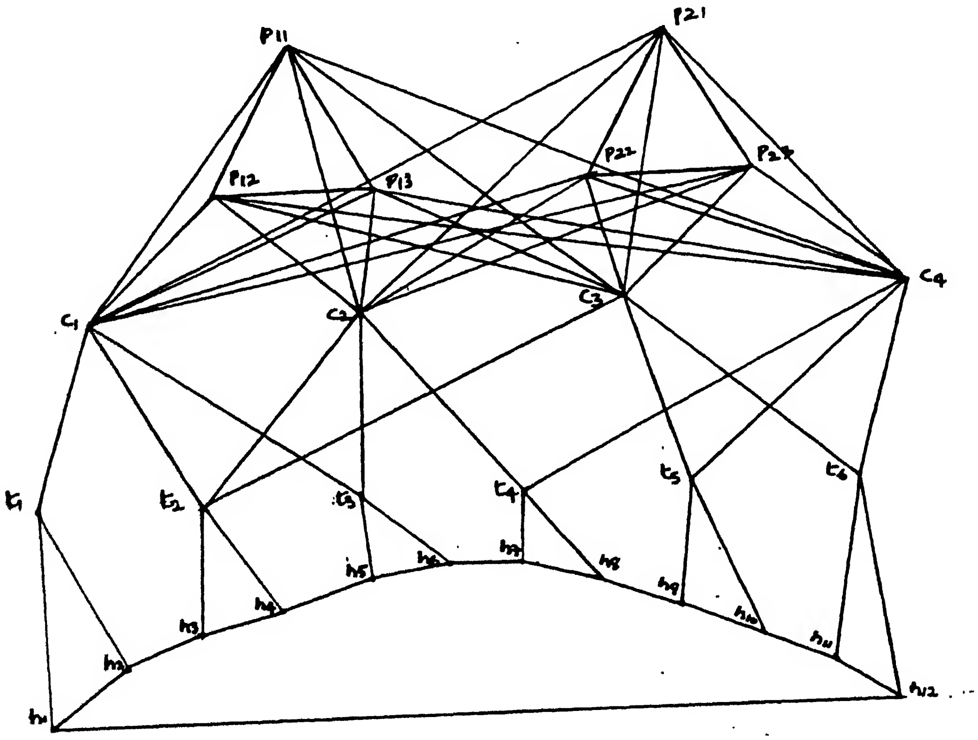


**Fig.2. Reduction for 3-RCCSP**

## 2.5 Connected Regular Subgraph Problem

The no. of components constraint will make this problem intractable even when the bound is the upper limit.

Instance: Undirected graph $G=(V,E)$, Where '$V$' is the set of vertices and '$E$' the set of edges between them.

Question: Does $G$ contain a connected *R-regular subgraph* $G'=(V,E')$ where $E' \subseteq E$?

**Theorem:** Connected $R$-Regular Subgraph Problem is $NP$-Complete, for even $R$.

**Proof:** (Transformation from Hamiltonian Cycle Problem).

Hamiltonian Cycle Problem: Given an Undirected graph $G=(V,E)$, Where '$V$' is the set of vertices and '$E$' the set of edges between them, Does there exist a cycle which visits each node exactly once?

We will show that HCP is reducible to CRSP.

Given any instance of HCP we define an instance of CRSP as follows.

Let the given graph $G=(V,E)$, where $|V|=n$ and the vertices are $v_1,...v_n$. The Instance of CRSP is defined as follows: For each node $v_i \in V$, we create a gadget with $R+1$ nodes with valency $R-2$ i.e. it requires $R-2$ edges from outside to become $R-regular$. we add *R-2* edges from this node $v_i$ to the gadget. Now, in any connected $R-regular$ subgraph of this graph all the edges in the gadget along with the edges between the gadgets and the corresponding $v_i$ vertices should be present. Since there aren't any connections between gadgets, we can get a connected *subgraph* if and only if there is a connected 2-regular graph (Hamiltonian Cycle) in the given HCP instance.

We prove the existence of such gadgets, for $R$ even.

First create a complete graph of $R+1$ nodes(each node has degree $R$). Now, delete $(R-2)/2$ edges( $R$ should be even in order this to be an integer) from this graph such that no node has lost more than one edge (for example, if the nodes are numbered $v_1,...,v_{(R+1)}$, delete all edges $(v_{2i},v_{(2i+1)}) for i = 1,...,((R-2)/2))$. It can

be clearly seen that in this graph three nodes are connected to the remaining $R$ nodes and all other nodes are connected to $R - 1$ nodes.

Hence, it is easy to see that Connected $R$–*Regular* (for $R$ even) Subgraph Problem is $NP$-Complete.

## 2.6 Regular and Component Size Constrained Subgraph Problem

Instance: Undirected graph $G = (V, E)$, Where '$V$' is the set of vertices and '$E$' the set of edges between them and an integer $K \leq |V|$, that fixes the upper bound on the no. of nodes in any component in the *subgraph.*

Question: Does $G$ contain a 3-*Regular subgraph* $G'= (V,E')$ where $E' \subseteq E$ such that the number of nodes in any component in the *subgraph* is less than or equal to $K$?

**Theorem:** 3-RCSCSP is $NP$-Complete.

**Proof:** (Transformation from Exact Cover By 3-Sets).

Exact Cover By 3-Sets (XS3): Given a family $S = \{c_1, ..., c_s\}$ of three element subsets of a set $T = \{t_1, ..., t_{3m}\}$, Does there exist a sub family $S' \subseteq S$ of pairwise disjoint sets such that $\bigcup c = T$ ?

We will show that XS3 is reducible to 3-RCSCSP.

Given any instance of XS3 we define an instance of RCSCSP as follows (Ref. Fig. 3).

$V = S \cup T \cup P$, Where $S = \{c_i/c_i \in S\}$, $\mid S \mid= n$.

$T = \{t_i/t_i \in T\}$, $\mid T \mid= 3m$.

$P = \{p_{ij}/(i = 1..., n-m)(j = 1, 2, 3)\}$, $\mid P \mid= 3(n-m)$.

$E = ST \cup SP \cup PP \cup TT$, Where $ST = \{(c, t) : t \in c \in S\}$

$SP = \{(c, p) : c \in S, p \in S\}$

$PP = \{(p_{i1}, p_{i2}), (p_{i2}, p_{i3}), (p_{i3}, p_{i1}) : p_i \in P\}$.

$TT = \{(t_i, t_j) : t_i \in T, t_j \in T, i \neq j\}$

/* $PP$ : $p_{i1}, p_{i2}, p_{i3}$ form a triangle. Hence n-m triangles */

/* $SP$: n-m triangles connected to all c nodes */

/* $TT$: All $t_i$'s are connected among themselves forming a 3m-complete graph */

$K = 4$.

Now, the graph $G = (V, E)$ will have a 3-*regular subgraph* with no component having more than $K$ nodes, if and only if the given XS3 instance has an exact cover.

When the given XS3 has a solution, it should comprise of 'm' three element subsets which should cover the '3m' elements, the corresponding 3-RCSCSP will have a solution where the vertices corresponding to chosen three element subsets will be connected to the three elements present in them and other subsets will be connected to one triangle each. Thus all components will have size 4 which is a solution. Before showing that the converse of this is also true, let us first note that a 4-node 3-regular graph is a complete graph. So, the solution subgraph consists of 'n' complete graphs of four nodes each. Since there are no edges between the p nodes and t nodes, no component can have both types of nodes present. Also when ever a p node is present in a component there should at least be one c node present. It can be seen that in fact not more than one c node can not be present in any component as there are no edges between the c nodes. Thus the 3(n-m) p nodes require (n-m) c nodes and the remaining c nodes, which are not connected among themselves should be connected to three t nodes each. Thus the subset vertices connected to to the t vertices exactly cover the set '$T$'.

Now, the reduction is shown to work and the transformation is *polynomial* time computable and since XS3 is $NP$-Complete 3-RCSCSP will also be $NP$-Complete.

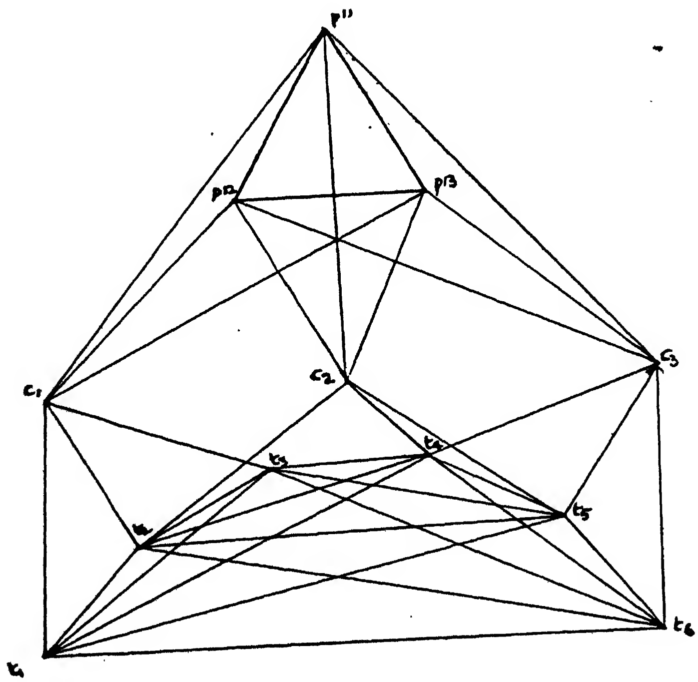

**Fig.3. Reduction for 3-RCSCSP**

The component size constraint makes this problem intractable even when the bound is the lower limit.

Instance: Undirected graph G=(V,E), and an integer $K \leq |V|$ that fixes the lower bound on the no. of nodes in any component in the subgraph.

Question: Does G contain an R-regular subgraph G' such that the no. of nodes in any component is greater than or equal to K?

**Theorem:** R-RCSCSP is NP-Complete for R even and $K = 6 * (R + 1)$.

**Proof:** (Transformation from Partition into Hamiltonian Subgraphs)

Partition into Hamiltonian Subgraphs: Given an Undirected graph $G = (V, E)$, Where '$V$' is the set of vertices and '$E$' the set of edges between them, Does there exist a partion of $V$ into disjoint sets $V_1, V_2, \ldots, V_p$, for some p, such that each $V_i$ contains at least six vertices and induces a subgraph of G that contains a Hamiltonian circuit?

The reduction of Partition into Hamiltonian Subgraphs to R-RCSCSP is exactly similar to that of HCP to CRSP given in page 20 and $K = 6 * (R + 1)$.

Here, in any component at least 6 gadjets should be present as the size should be greater than or equal to 6(R+1). So, the given PHS instance should have at least 6 nodes present in any component.

Now, the reduction is shown to work and the transformation is *polynomial* time computable and since PHS is $NP$-Complete R-RCSCSP will also be $NP$-Complete.

# Chapter 3

# Concluding Remarks

The techniques used previously to show the $NP$-Completeness can perhaps be extended to weighted graphs. Here, the *local* constraint can be generalized to total weight at each node. Whereupon the degree constraint becomes a special case (when all edges have weight equal to 1) of this. We illustrate some easy cases of the weighted graphs below.

Total Weight Constrained Subgraph Problem

For weighted graphs, we will get an $NP$-Complete problem even with just a *global* constraint, namely, the total weight of the graph is equal to '$B$'.

Instance: A weighted graph $G = (V, E)$ and an integer $B$.

Question: Does $G$ contain a *subgraph* $G' = (V, E')$ where $E' \subseteq E$ such that the total weight of the graph $G'$ is exactly equal to $B$?

**Theorem:** TWCSP is $NP$-Complete.

**Proof:** (Transformation from PARTITION).

PARTITION: Given a finite set $A$ and a size $S(a) \in Z^+$ for each $a \in A$, Is there a subset $A' \subset A$ such that

$$\sum_{a \in A'} S(a) = \sum_{a \in A - A'} S(a)?$$

We will show that PARTITION is reducible to TWCSP.

Given any instance of PARTITION we define an instance of TWCSP as follows.

$$B = \sum_{a \in A} S(a)/2$$

Create a sufficiently big random graph $G = (V, E)$ where $\mid E \mid = \mid A \mid$ and then assign weights for the edges from the set $A$ such that each $a \in A$ is assigned to only one $e \in E$ and viceversa.

It is easy to see that this random graph will have a *subgraph* whose total weight is equal to $B$ iff the given instance can be partitioned into two equal halves.

Hence TWCSP is $NP$-Complete.

Now, it can be seen that the other two variants, namely, the total weight $\leq B$ and the total weight $\geq B$ are trivially in $P$. Further, these two problems can be solved efficiently even with addition of the constraint of the form ' No. of components $\leq K$'.

Towards the characterization of natural properties that make a problem hard we have considered two kinds, namely, local and global constraints satisfied by the solution subgraph. We showed that certain combinations of these constraints make a problem intractable and certain others make the problem easy. We have obtained some results more in the nature of data. We do see the scope of a more elaborate and more extensive study. Clearly, the notions of local and global constraints are natural and it is useful to view problems in these terms. However, a more detailed analysis of locality, globality and their interaction is necessary before one can say if it is possible to build a metatheory based on these notions to explain the $NP$-Completeness phenomenon. Also, some work exists that views the complexity of subgraph problems in terms of the minimal logics of finite structures necessary to express the subgraph in question [AB91]. It will be intersting to know if some connection exists between the logic approach and the approach taken in this thesis.

# Bibliography


[AB91] V. Arvind AND S. Biswas [1991], "Edge-deletion graph problems with first-order expressible subgraph properties," International Journal of Foundations of Computer Science Vol 2, No. 2, 83-99.

[C71] Stephen A. Cook [1971], "The complexity of theorem proving procedures, " Proceedings of 3rd Annual ACM Symposium on Theory of Computing, Association for Computing Machinery, New York, 151-158.

[J84] David S. Johnson [1984],"NP-Completeness column: An ongoing guide, " Journal of Algorithms 5, 147-157.

[J85] David S. Johnson [1985],"NP-Completeness column: An ongoing guide, " Journal of Algorithms 6, 145-159.

[GJ79] Michael R. Garey AND David S. Johnson [1979], "Computers and intractability A guide to the theory of NP-Completeness," W.H.Freeman and Company, San Francisco.

[GM84] M.Gondra AND M.Minoux [1984], "Graphs and Algorithms," Wiley-Interscience Publication.